Journal of Applied Analysis and Computation
Volume 17, Number 1, February 2027, 524–539
Website:http://www.jaac-online.com
DOI:10.11948/20250257

# A NOISE-TOLERANT AND FIXED-TIME ZEROING NEURAL NETWORK MODEL FOR SOLVING MULTI-LINEAR $\mathcal{M}$-TENSOR EQUATIONS

**Ting-Jia Liu[1,†], Mengyao Li[2], Ruijuan Zhao[2] and Shu-Xin Miao[1]**

**Abstract** Zeroing neural network (ZNN) models are effective methods for solving multi-linear systems. To improve the performance of ZNN models, a new ZNN model is proposed for solving multi-linear systems with $\mathcal{M}$-tensor in this paper. The proposed ZNN model incorporates with a nonlinear activation function. Convergence and robustness of the proposed ZNN model are analyzed. Theoretical results reveal that the proposed ZNN model converges within a fixed time and tolerates different types of noise. Numerical experiments illustrate that the proposed ZNN model is more powerful than some existing ZNN models for solving multi-linear systems with $\mathcal{M}$-tensor.

**Keywords** Multi-linear system, ZNN model, fixed-time convergence, noise-tolerant.

**MSC(2010)** 15A69, 65F15, 68W25.

## 1. Introduction

Multi-linear systems play an important role in numerical partial differential equations, data mining, tensor complementarity problems, transportation systems and control [3, 4, 15, 18, 29]. Due to the multi-linear term in the systems, the solvability conditions of multi-linear systems need to be carefully studied, there are some theoretical results on the solvability of multi-linear systems [1, 3, 12, 16]. In particular, Ding and Wei [3] show that the multi-linear systems have a unique positive solution if the coefficient tensor is a nonsingular $\mathcal{M}$-tensor (see Section 2 for its definition) and the right hand vector is a positive vector. Hence, we pay our attention to the multi-linear systems with $\mathcal{M}$-tensors in this paper.

Because of the wide application, numerical methods have been proposed for solving multi-linear systems, including the homotopy method [6], the tensor method [31], the Levenberg-Marquardt method [19], the block iterative methods [20], several classical iterative methods [3, 7, 13] and its preconditioned forms [14, 21, 30].

Besides the numerical methods mentioned above, neural network models are also efficient methods for solving multi-linear systems [24, 25, 27]. The neural network models, including the gradient-based neural network (GNN) models and the zeroing neural network (ZNN) models

†The corresponding author.
[1]College of Mathematics and Statistics, Northwest Normal University, Lanzhou, China
[2]School of Information Engineering, Lanzhou University of Finance and Economics, Lanzhou, China
Email: liutingjia1128@163.com(T.-J. Liu), lmy12581@163.com(M. Li), zhaobin7755382@163.com(R. Zhao), miaosx@nwnu.edu.cn(S.-X. Miao)

[34], have now been regarded as the powerful alternative methods for solving some engineering and mathematics problems in view of its natures of distributed-storage, high-speed parallel computing and the convenience of hardware implementation. Specifically, the GNN models and the ZNN models have been studied for solving the matrix inversion [10], matrix equations [33], programming problems [32], as well as multi-linear systems [24, 25]. As for solving multi-linear systems with $\mathcal{M}$-tensors, the continuous time neural network (CTNN) model and its modified versions [26], the modified Newton integration neural algorithm [9], the finite-time ZNN (FTZNN) model [24] and the recurrent neural networks [28] have been proposed recently. Moreover, the complex-valued ZNN models [27] and the discrete-time neural dynamics model [17] have also been presented to solve the time-varying multi-linear systems.

The convergence time and the robustness against noise are the key factors to measure the performance of the neural network models [9, 17]. For accelerating the convergence time of the neural network models, one usually introduces nonlinear activation functions in the models [5, 10, 27]. For example, when the weighted sign-bi-power activation function is introduced in the complex-valued ZNN models, the fixed-time convergence of the complex-valued ZNN models for solving the time-varying multi-linear systems can be achieved [27]. The control strategies and the nonlinear activation functions in the neural network models have both contributions to the characters of noise tolerance [9, 25].

For solving the multi-linear systems with $\mathcal{M}$-tensors, the models in [24] have the finite-time convergence, the models in [27] have the fixed-time convergence, and the model in [17] has the finite-time convergence and the noise tolerance. Best to our knowledge, there is no model that has both fixed-time convergence and noise tolerance for solving the multi-linear systems with $\mathcal{M}$-tensors. Hence, we propose a new ZNN model incorporates with a nonlinear activation function for solving multi-linear systems with $\mathcal{M}$-tensors, the main contributions of this study are listed below:

(1) Compared with existing neural network model [24], our suggested model has a shorter fixed-time convergence under some conditions. And compared with continuous time neural network model [26], the model retains noise-tolerant performance and the fixed-time convergence.

(2) Theoretical results show that the proposed model achieves fixed-time convergence both in an ideal surroundings and in the presence of two types of noise disturbances.

(3) Numerical experiments validate the theoretical results, demonstrate the effectiveness of the proposed model.

The rest parts of this paper are organized as follows. In Section 2, some preliminary results are described. The noise-tolerant zeroing neural network model with fixed-time convergence for online solution of the multi-linear systems with $\mathcal{M}$-tensors is proposed in Section 3. In Section 4, the convergence and robustness of the proposed model are verified. In Section 5, some numerical experiments are performed to illustrate the effectiveness of the proposed model. Conclusions of this paper are given in Section 6.

## 2. Preliminaries

In this section, we present some necessary preliminaries on the multi-linear system and the differential dynamic system.

Let $\mathbb{R}$ and $\mathbb{C}$ be the real field and complex field, respectively. The set of all $m$-order, $n$-dimensional real tensors is denoted by $\mathbb{R}^{[m,n]}$. $\mathbb{R}^n$ is the space composed of all $n$-tuples of real numbers, i.e., $\mathbb{R}^n = \{(x_1, x_2, \ldots, x_n) : x_i \in \mathbb{R}, i = 1, 2, \ldots, n\}$, where $\mathbb{R}$ denotes the set of real numbers. $\mathbb{C}^n$ is the space composed of all $n$-tuples of complex numbers, i.e., $\mathbb{C}^n = \{(c_1, c_2, \ldots, c_n) : c_i \in \mathbb{C}, i = 1, 2, \ldots, n\}$, where $\mathbb{C}$ denotes the set of complex numbers. $\mathbb{R}^{n\times n}$ is the set of all $n \times n$ real matrices. Formally, $\mathbb{R}^{n\times n}$ can be defined as $\mathbb{R}^{n\times n} = \{A = (a_{ij}) : a_{ij} \in \mathbb{R}, 1 \le i, j \le n\}$, where $A$ is an $n \times n$ matrix, $a_{ij}$ represents the element in the $i$th row and $j$th column of matrix $A$. In this paper, we consider the multi-linear system

$$\mathcal{A}x^{m-1} = b, \tag{2.1}$$

where $\mathcal{A} \in \mathbb{R}^{[m,n]}$ is an $\mathcal{M}$-tensor, $b$ and $x$ are vectors in $\mathbb{R}^n$, the $n$-dimensional vector $\mathcal{A}x^{m-1}$ is defined as [23]

$$(\mathcal{A}x^{m-1})_i = \sum_{i_2,\ldots,i_m=1}^{n} a_{ii_2,\ldots,i_m} x_{i_2} \ldots x_{i_m}, i = 1, 2, \ldots, n,$$

$x_i$ denotes the $i$th component of $x$.

**Definition 2.1.** [8] A pair $(\lambda, x) \in \mathbb{C} \times (\mathbb{C}^n \setminus 0)$ is called an eigenpair of $\mathcal{A} \in \mathbb{R}^{[m,n]}$, if it satisfies the system of equations

$$\mathcal{A}x^{m-1} = \lambda x^{[m-1]},$$

where $x^{[m-1]} = [x_1^{[m-1]}, x_2^{[m-1]}, \ldots, x_n^{[m-1]}]^\top$. $\rho(\mathcal{A}) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } \mathcal{A}\}$ denotes the spectral radius of tensor $\mathcal{A}$.

**Definition 2.2.** [2, 3] If there exists a nonnegative tensor $\mathcal{B}$, the unit tensor $\mathcal{I}$ and the real value $s$ satisfy $\mathcal{A} = s\mathcal{I} - \mathcal{B}$, where $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{[m,n]}$, then the tensor $\mathcal{A}$ is called an $\mathcal{M}$-tensor when $s \ge \rho(\mathcal{B})$. Furthermore, $\mathcal{A}$ is called a nonsingular $\mathcal{M}$-tensor if $s > \rho(\mathcal{B})$.

The tensor $\mathcal{A} \in \mathbb{R}^{[m,n]}$ is called symmetric if its elements are invariant under any permutation of indices $i_1 i_2 \ldots i_m$ [23]. If $\mathcal{A} \in \mathbb{R}^{[m,n]}$ is a symmetric tensor, then for any $x \in \mathbb{R}^n$, $\mathcal{A}x^{m-2}$ is an $n \times n$ matrix with elements

$$(\mathcal{A}x^{m-2})_{ij} = \sum_{i_3,\ldots,i_m}^{n} a_{iji_3,\ldots,i_m} x_{i_3} \ldots x_{i_m}, i, j = 1, 2, \ldots, n.$$

For a general tensor $\mathcal{A} \in \mathbb{R}^{[m,n]}$, one can partially symmetrize $\mathcal{A}$ with respect to indices $i_2, \ldots, i_m$ to the partial symmetrized tensor $\hat{\mathcal{A}} = (\hat{a}_{ii_2\ldots i_m}) \in \mathbb{R}^{[m,n]}$ [6], where

$$\hat{a}_{ii_2\ldots i_m} = \frac{1}{(m-1)!} \sum_{\pi} a_{i_1\pi(i_2\ldots i_m)},$$

and the sum is over all the permutations $\pi(i_2 \ldots i_m)$. For any $\mathcal{A} \in \mathbb{R}^{[m,n]}$, we can get a partially symmetrized tensor $\hat{\mathcal{A}} \in \mathbb{R}^{[m,n]}$ such that $\mathcal{A}x^{m-1} = \hat{\mathcal{A}}x^{m-1}$, by an averaging procedure.

**Lemma 2.1.** [6] *If $\mathcal{A} \in \mathbb{R}^{[m,n]}$ is a nonsingular $\mathcal{M}$-tensor, so is $\hat{\mathcal{A}}$.*

**Lemma 2.2.** [3] *Let $\mathcal{A} \in \mathbb{R}^{[m,n]}$ is a nonsingular $\mathcal{M}$-tensor. If there exists a vector $x \in \mathbb{R}^n$ such that $\mathcal{A}x^{m-1} > 0$, then matrix $\hat{\mathcal{A}}x^{m-2}$ is a nonsingular symmetric M-matrix, and thus a positive definite matrix.*

In what follows of this section, we recall some basic definition of the differential dynamic system

$$\dot{x}(t) = g(x(t)),\ x(0) = x_0,\ t \in [0, \infty)\,, \tag{2.2}$$

where $x(t) \in \mathbb{R}^n$ is the system state, and $g(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear function.

**Definition 2.3.** [11] Let $x_*(t) \in \mathbb{R}^n$, then it is called an equilibrium point of the equation (2.2) if $g(x_*(t)) = 0$.

Assuming that the origin $x(t) = 0 \in \mathbb{R}^n$ is an equilibrium solution of the system (2.2).

**Definition 2.4.** [22] The origin of system (2.2) is globally finite-time stable if it is globally asymptotically stable and there is the setting-time function $T : \mathbb{R}^n \to \mathbb{R}_+ \bigcup\{0\}$ which makes every solution $x(t, x_0)$ reaches the equilibria at some finite time moment, i.e. $x(t, x_0) = 0, \forall\ t \geq T(x_0)$.

**Definition 2.5.** [22] The origin of system (2.2) is fixed-time stable if it is globally finite-time stable and the setting-time function $T$ is bounded, i.e., $\exists\ T_{max} > 0 : T_{max} \geq T_{x_0}, \forall\ x_0 \in \mathbb{R}^n$.

## 3. The proposed model for solving multi-linear system

In this section, after reviewing the ZNN model for solving the multi-linear system (2.1), we propose a noise-tolerant zeroing neural network model with the fixed-time convergence (FTC-NTZNN model hereafter) to solve the multi-linear system (2.1).

The standard ZNN model for solving the multi-linear system (2.1) can be defined by three steps. Firstly, we define the following error function $E(t)$ of the multi-linear system (2.1)

$$E(t) = \mathcal{A}x(t)^{m-1} - b. \tag{3.1}$$

Secondly, the standard zeroing design formula [26, 34]

$$\dot{E}(t) = \frac{\mathrm{d}E(t)}{\mathrm{d}t} = -\gamma(E(t)) \tag{3.2}$$

with parameter $\gamma > 0$ is given to make the error function $E(t)$ converges to 0. Finally, assume that $\hat{\mathcal{A}}x(t)^{m-2}$ is nonsingular, substituting (3.1) into (3.2) gives the standard ZNN model

$$\dot{x}(t) = -\frac{\gamma}{m-1}(\hat{\mathcal{A}}x(t)^{m-2})^{-1}(\mathcal{A}x(t)^{m-1} - b) \tag{3.3}$$

for solving (2.1).

It is well known that the convergence performance of the ZNN model can be improved by incorporating some activation functions in the ZNN model [24, 26]. Hence, when the array activation function $\psi(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ is introduced in (3.3), we then obtain the activation function version ZNN model

$$\dot{x}(t) = -\frac{\gamma}{m-1}(\hat{\mathcal{A}}x(t)^{m-2})^{-1}\psi(\mathcal{A}x(t)^{m-1} - b). \tag{3.4}$$

The activation function version ZNN model (3.4) with different activation functions give different models, for example, when the linear array activation function $\psi_1(y) = y$ for $y \in \mathbb{R}^n$ is used, the CTNN model has been proposed for solving (2.1) [26], when the sign-bi-power array activation function $\psi_2(y) = \frac{1}{2}\mathrm{sig}^r(y) + \frac{1}{2}\mathrm{sig}^{\frac{1}{r}}(y)$ for $y \in \mathbb{R}^n$ is used, the FTZNN model [24] is established for solving (2.1), here $r > 0$ and $\mathrm{sig}^r(y)$ is defined as

$$\mathrm{sig}^r(y) = \begin{cases} |y|^r, & \text{if } \ y > 0, \\ 0, & \text{if } \ y = 0, \\ -|y|^r, & \text{if } \ y < 0. \end{cases}$$

Theoretical results as well as numerical experiments have demonstrated that the FTZNN model [24] outperforms the CTNN model [26]. Note that the FTZNN model [24] has the character of the finite-time convergence, but may has not tolerances noise perturbations.

To improve the convergence time and tolerates the noise perturbations of the activation function version ZNN model (3.4), we propose the following FTCNTZNN model

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = -\frac{\gamma}{m-1}(\hat{\mathcal{A}}x(t)^{m-2})^{-1}\psi_3(\mathcal{A}x(t)^{m-1} - b) \tag{3.5}$$

for solving (2.1), where the array activation function $\psi_3(y)$ is given by

$$\psi_3(y) = a_1^q \exp(q|y|^p)|y|^{1-p}\mathrm{sgn(y)/(pq) + a_2y + a_3sgn(y), \ \ y \in \mathbb{R}^n}$$

with parameters $a_1 > 0, a_2 \geq 0, a_3 \geq 0, 0 < p < 1, q > 0$, and sign function $\mathrm{sgn}(\cdot)$. Note that the array activation function $\psi_3(x)$ is used in [5] to accelerates the convergent rate of the ZNN model for solving the time-varying Sylvester equation.

## 4. Convergence and robustness of the FTCNTZNN model

In this section, the convergence and robustness of the FTCNTZNN model (3.5) will be theoretically analyzed.

### 4.1. Convergence analysis

Let $x_*$ be the solution of the multi-linear system (2.1), and $\Omega(x_*, \delta) = \{x \in \mathbb{R}^n | \|x - x_*\|_2 \leq \delta, \delta > 0\}$ be a neighborhood of $x_*$. According to Lemma 2.2, the matrix $\hat{\mathcal{A}}x_*^{m-2}$ is nonsingular when $\mathcal{A}$ is an $\mathcal{M}$-tensor, so that $\hat{\mathcal{A}}x(t)^{m-2}$ is nonsingular for any $x(t) \in \Omega(x_*, \delta)$.

First, we present the global asymptotic convergence theory of the FTCNTZNN model (3.5) for solving the multi-linear system (2.1).

**Theorem 4.1.** *Let $x_*$ be the solution of the $\mathcal{M}$-tensor multi-linear system (2.1). Assuming that the FTCNTZNN model (3.5) with the activation function $\psi_3(y)$ is used for solving the $\mathcal{M}$-tensor multi-linear system (2.1). Then the state vector $x(t)$ of the FTCNTZNN model (3.5) globally asymptotically converges to the theoretical solution $x_*$ for any initial value $x(0) \in \Omega(x_*, \delta)$.*

**Proof.** Let

$$L_E(t) = \frac{1}{2}\|E(t)\|_2^2 = \frac{1}{2}E^\top(t)E(t)$$

be the candidate Lyapunov function. Then the derivative of $L_E(t)$ with respect to $t$ is

$$\dot{L}_E(t) = \frac{\mathrm{d}L_E(t)}{\mathrm{d}t}$$

$$
\begin{aligned}
&= \frac{1}{2}(\dot{E}(t)^\top E(t) + E(t)^\top \dot{E}(t)) \\
&= E(t)^\top \dot{E}(t) \\
&= -\gamma E(t)^\top \psi_3(E(t)) \\
&= -\gamma \sum_{j=1}^{n} e_j(t)\psi_3(e_j(t)),
\end{aligned}
$$

where $e_j(t)$ is the $j$th element of the error function $E(t)$.

As the activation function $\psi_3(\cdot)$ is a monotonically increasing odd function, then we have $\psi_3(-e_j(t)) = -\psi_3(e_j(t))$ and

$$
\psi_3(e_j(t)) \begin{cases} > 0, & \text{if } e_j(t) > 0, \\ = 0, & \text{if } e_j(t) = 0, \\ < 0, & \text{if } e_j(t) < 0. \end{cases}
$$

Therefore,

$$
e_j(t)\psi_3(e_j(t)) \begin{cases} > 0, & \text{if } e_j(t) \neq 0, \\ = 0, & \text{otherwise.} \end{cases}
$$

Note that $\gamma > 0$, we have

$$
\dot{L}_E(t) \begin{cases} < 0, & \text{if } E(x(t)) \neq 0, \\ = 0, & \text{if } E(x(t)) = 0. \end{cases}
$$

According to the Lyapunov stability theory, the state vector $x(t)$ globally asymptotically convergent to the theoretical solution $x_*$. □

Next, we give the fixed-time convergence theory of the FTCNTZNN model (3.5) for solving the multi-linear system (2.1).

**Theorem 4.2.** *If the conditions of Theorem 4.1 are satisfied, then the state vector $x(t)$ of the FTCNTZNN model (3.5) with any initial value $x(0) \in \Omega(x_*, \delta)$ converges to the theoretical solution $x_*$ within the fixed time*

$$
t(x(0)) \leq \frac{1}{\gamma a_1^q} = T_{\max}.
$$

**Proof.** It follows from $\dot{E}(t) = -\gamma\psi_3(E(t))$ that

$$
\dot{e_j}(t) = -\gamma\psi_3(e_j(t)),\ j = 1, 2, \ldots, n,
$$

where $e_j(t)$ and $\dot{e_j}(t)$ are the $j$th element of the error function $E(t)$ and its derivative$\dot{E}(t)$, respectively. Define the Lyapunov function

$$
L(t) = |e_j(t)| \geq 0,
$$

then we have its time derivative

$$
\dot{L}(t) = \dot{e_j}(t)\mathrm{sgn(e_j(t))}
$$

$$
\begin{aligned}
&= -\gamma[a_1^q \exp(q|e_j(t)|^p)|e_j(t)|^{1-p}\mathrm{sgn(e_j(t))/(pq) + a_2e_j(t) + a_3sgn(e_j(t))]sgn(e_j(t))}\\
&= -\gamma[a_1^q \exp(q|e_j(t)|^p)|e_j(t)|^{1-p}/(pq) + a_2|e_j(t)| + a_3]\\
&\leq -\gamma a_1^q \exp(q|e_j(t)|^p)|e_j(t)|^{1-p}/(pq)\\
&= -\gamma a_1^q \exp(qL^p(t))L^{1-p}(t)/(pq).
\end{aligned}
$$

As $\dot{L}(t) \leq -\gamma a_1^q \exp(qL^p(t))L^{1-p}(t)/(pq) \leq 0$, by the Lyapunov stability theory, the FTCNTZNN model (3.5) globally converges to zero over time.

The differential form of $\dot{L}(t) \leq -\gamma a_1^q \exp(qL^p(t))L^{1-p}(t)/(pq)$ is

$$\gamma a_1^q \mathrm{d}t \leq -pqL^{p-1}(t)\exp(-qL^p(t))\mathrm{d}L(t). \tag{4.1}$$

Integrating (4.1) gives

$$\int_0^{t_j} \gamma a_1^q \mathrm{d}t \leq -\int_{L_0}^0 pqL^{p-1}(t)\exp(-qL^p(t))\mathrm{d}L(t),$$

that is

$$\gamma a_1^q t\big|_0^{t_j} \leq \exp(-qL^p(t))\big|_{L(0)}^0.$$

After some calculation, we have

$$t_j \leq \frac{1-\exp(-qL^p(0))}{\gamma a_1^q}.$$

Note that $\frac{1-\exp(-qL^p(0))}{\gamma a_1^q} \leq \frac{1}{\gamma a_1^q}$. Hence, the upper bound of the convergence time of the FTCNTZNN model (3.5) is

$$t(x(0)) = \max\{t_j\} \leq \frac{1}{\gamma a_1^q} = T_{\max}.$$

The proof is completed. □

## 4.2. Robustness analysis

In the implementation of the FTCNTZNN model (3.5), noise is unavoidable. Let us consider the noise-perturbed FTCNTZNN model

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = -\frac{\gamma}{m-1}(\hat{\mathcal{A}}x(t)^{m-2})^{-1}\psi_3(\mathcal{A}x(t)^{m-1} - b) + Z(t) \tag{4.2}$$

of the FTCNTZNN model (3.5), where $Z(t)$ is the noise. Two types of noise are considered in what follows of this subsection, one is the dynamic bounded vanishing noise $Z(t)$, its $j$th element satisfies $|z_j(t)| \leq \delta_1|e_j(t)|$, $\delta_1 \in (0,\infty)$, and the other is dynamic bounded non-vanishing noise $Z(t)$, its $j$th element satisfies $|z_j(t)| \leq \delta_1$, $\delta_1 \in (0,\infty)$.

For the dynamic bounded vanishing noise $Z(t)$, we have the following convergence result.

**Theorem 4.3.** *Assume that the noise-perturbed FTCNTZNN model (4.2) with the dynamic bounded vanishing noise $Z(t)$ is used to solve the the $\mathcal{M}$-tensor multi-linear system (2.1). If $\gamma a_2 \geq \delta_1$, then the state vector $x(t)$ of the model (4.2) starting from any initial value $x(0)$ converges to the theoretical solution $x_*$ of (2.1) in the fixed-time $T_{\max}$.*

**Proof.** The noise-perturbed FTCNTZNN model (4.2) can be rewritten as

$$\dot{E}(t) = -\gamma\psi_3(E(t)) + Z(t). \tag{4.3}$$

The $j$th subsystem of (4.3) is

$$\dot{e}_j(t) = -\gamma\psi_3(e_j(t)) + z_j(t), \tag{4.4}$$

where $j = 1, 2, \ldots, n$.

Let $u(t) = |e_j(t)|^2 \geq 0$ be the Lyapunov function, its derivative with respect to time $t$ is

$$\dot{u}(t) = 2e_j(t)\dot{e_j}(t) = 2e_j(t)(-\gamma\psi_3(e_j(t)) + z_j(t)). \tag{4.5}$$

Substituting $\psi_3$ and (4.4) into (4.5), and note that $Z(t)$ is the dynamic bounded vanishing noise, we have

$$\begin{aligned}
\dot{u}(t) &= -2\gamma a_1^q exp(q|e_j(t)|^p)|e_j(t)|^{2-p}/(pq) - 2\gamma a_3|e_j(t)| + 2(e_j(t)z_j(t) - \gamma a_2|e_j(t)|^2) \\
&\leq -2\gamma a_1^q \exp(q|e_j(t)|^p)|e_j(t)|^{2-p}/(pq) + 2(\delta_1|e_j(t)|^2 - \gamma a_2|e_j(t)|^2) \\
&\leq -2\gamma a_1^q \exp(q|e_j(t)|^p)|e_j(t)|^{2-p}/(pq) \\
&= -\gamma a_1^q \exp(qu^{\frac{p}{2}}(t))u^{\frac{2-p}{2}}(t)/(pq/2) \\
&\leq 0.
\end{aligned}$$

From the Lyapunov theory, we known that the noise-perturbed FTCNTZNN model (4.2) with the dynamic bounded vanishing noise $Z(t)$ globally converges to zero. Moreover, similar to the proof of Theorem 4.2, we can get that the noise-perturbed FTCNTZNN model (4.2) with the dynamic bounded vanishing noise $Z(t)$ has the fixed-time convergence, and the the upper bound of the convergence time is $T_{\max}$. □

As for the dynamic bounded non-vanishing noise $Z(t)$, we obtain the following convergence result.

**Theorem 4.4.** *Assume that the noise-perturbed FTCNTZNN model (4.2) with the dynamic bounded non-vanishing noise $Z(t)$ is used to solve the $\mathcal{M}$-tensor multi-linear system (2.1). If $\gamma a_3 \geq \delta_1$, then the state vector $x(t)$ of the model (4.2) starting from any initial value $x(0)$ converges to the theoretical solution $x_*$ of (2.1) in the fixed-time $T_{\max}$.*

**Proof.** Similar to the proof of Theorem 4.3, for the Lyapunov function as $u(t) = |e_j(t)|^2 \geq 0$ and its derivative $\dot{u}(t)$, note that $Z(t)$ is the dynamic bounded non-vanishing noise, we then have

$$\begin{aligned}
\dot{u}(t) &= -2\gamma a_1^q \exp(q|e_j(t)|^p)|e_j(t)|^{2-p}/(pq) - 2\gamma a_2|e_j(t)|^2 + 2(e_j(t)y_j(t) - \gamma a_3|e_j(t)|) \\
&\leq -2\gamma a_1^q \exp(q|e_j(t)|^p)|e_j(t)|^{2-p}/(pq) + 2(\delta_1|e_j(t)| - \gamma a_3|e_j(t)|) \\
&\leq -2\gamma a_1^q \exp(q|e_j(t)|^p)|e_j(t)|^{2-p}/(pq) \\
&= -\gamma a_1^q \exp(qu^{\frac{p}{2}}(t))u^{\frac{2-p}{2}}(t)/(pq/2) \\
&\leq 0.
\end{aligned}$$

The rest part of the proof is similar to the proof of Theorem 4.3, here we omitted. □

## 5. Numerical experiments

This section includes two examples aimed at illustrate the convergence and robustness performance of the FTCNTZNN models (3.5) and (4.2). All simulations were executed in MATLAB R2018a on a PC featuring an Intel(R) Core(TM) i5-8265U processor with a clock speed of 1.80GHz, 8GB of memory, and running Windows 11. The numerical solutions for equations (3.5) and (4.2), which describe the time-invariant multi-linear system, were obtained using the ode45 solver integrated within MATLAB.

The first numerical example will demonstrate the superiority of the FTCNTZNN model (3.5) for solving multi-linear nonsingular $\mathcal{M}$-tensor equation (2.1). It will illustrate how FTCNTZNN model (3.5) effectively addresses the nonsingular $\mathcal{M}$-tensor equation (2.1), showing its superiority over two models: A neural network utilizing a linear activation function (i.e., the CTNN model [26]) and the zeroing neural network with finite-time convergence based extended activation function (i.e., the FTZNN model [24]). For the second example will focus on multi-linear system with nonsingular $\mathcal{M}$-tensor. It will emphasize the strong robustness and the fastest convergence time of the FTCNTZNN model. Without loss of generality, the common model parameters are set as the following two cases:

**Case 1.** $a_1 = 2, a_2 = a_3 = 1, p = \frac{1}{6}, q = 3, r = \frac{1}{2}, \gamma = 40.$

**Case 2.** $a_1 = \frac{1}{2}, a_2 = a_3 = 1, p = \frac{1}{6}, q = \frac{1}{5}, r = \frac{1}{2}, \gamma = 2.$

**Example 5.1.** Consider the following second-order ordinary differential equation

$$\frac{d^2x(t)}{dt^2} = -\frac{GM}{x(t)^2}, \qquad t \in (0,1),$$

with Dirichlet's boundary conditions

$$x(0) = c_0, \qquad x(1) = c_1,$$

where $G \approx 6.67 \times 10^{-11}\,\mathrm{N}\cdot\mathrm{m}^2/\mathrm{kg}^2$ is the gravitational constant and $M \approx 5.98 \times 10^{24}\,\mathrm{kg}$ is the mass of the Earth. When $c_0, c_1 > 0$, this equation models the motion of a particle under gravitational attraction.

After the discretization, we obtain the following system of polynomial equations:

$$\begin{cases} x_1^3 = c_0^3, \\ 2x_i^3 - x_i^2 x_{i-1} - x_i^2 x_{i+1} = \dfrac{GM}{(n-1)^2}, \qquad i = 2, 3, \ldots, n-1, \\ x_n^3 = c_1^3. \end{cases}$$

This system can be expressed compactly as a multi-linear equation $\mathcal{A}x^3 = b$, where $\mathcal{A} = (a_{ijkl})$ is a fourth-order coefficient tensor, $x^3$ denotes the Kronecker product of the vector $x$, and $b$ is a given vector. The nonzero entries of $\mathcal{A}$ are

$$\begin{cases} a_{1111} = a_{nnnn} = 1, \\ a_{iiii} = 2, \qquad i = 2, 3, \ldots, n-1, \\ a_{i(i-1)ii} = a_{ii(i-1)i} = a_{iii(i-1)} = -\dfrac{1}{3}, \\ a_{i(i+1)ii} = a_{ii(i+1)i} = a_{iii(i+1)} = -\dfrac{1}{3}, \end{cases}$$

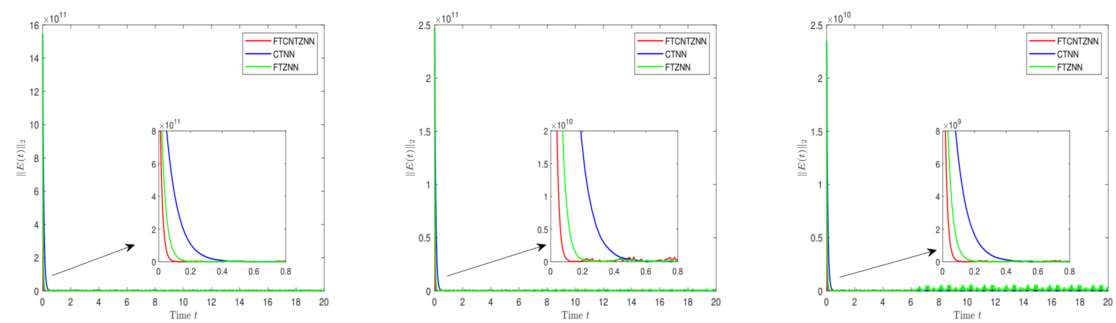
and the right-hand side vector $b = (b_i)$ is

$$\begin{cases} b_1 = c_0^3, \\ b_i = \dfrac{GM}{(n-1)^2}, \qquad i = 2, 3, \ldots, n-1, \\ b_n = c_1^3. \end{cases}$$

It can be verified that the coefficient tensor $\mathcal{A}$ is a nonsingular $\mathcal{M}$-tensor. Hence, the discretized problem is amenable to solution by the zeroing neural network models introduced in this work. Assume that the particle is located on the surface of earth, so that we choose $c_0, c_1 \approx 6370km$. Then the initial state can be set as $x(0) = c_0 * [1, 1, \cdots, 1]^\top$.

This example investigates the convergence performance of three zeroing neural network methods (FTCNTZNN, CTNN, and FTZNN) for solving nonsingular $\mathcal{M}$-tensors of order $m = 4$ with varying dimensions $n = 10, 20, 50$. The error norm $\|E(t)\|_2$ is plotted against time in each figure.

Firstly, experimental results show that as the tensor dimension $n$ increases from 10 to 50, the problem scale and computational complexity grow significantly, leading to larger initial error norms for all three methods. Across all test cases, the FTCNTZNN method exhibits superior convergence behavior, with faster and more stable error reduction. While CTNN and FTZNN also converge, they do so more slowly in higher dimensions and maintain slightly higher steady-state errors compared to FTCNTZNN. It is noteworthy that as $n$ increases, the number of time required for convergence also extends, consistent with the increased computational burden associated with higher-dimensional problems. Overall, FTCNTZNN demonstrates better numerical stability and convergence efficiency for medium to high-dimensional nonsingular $\mathcal{M}$-tensor systems, making it a more suitable choice for large-scale tensor computations.

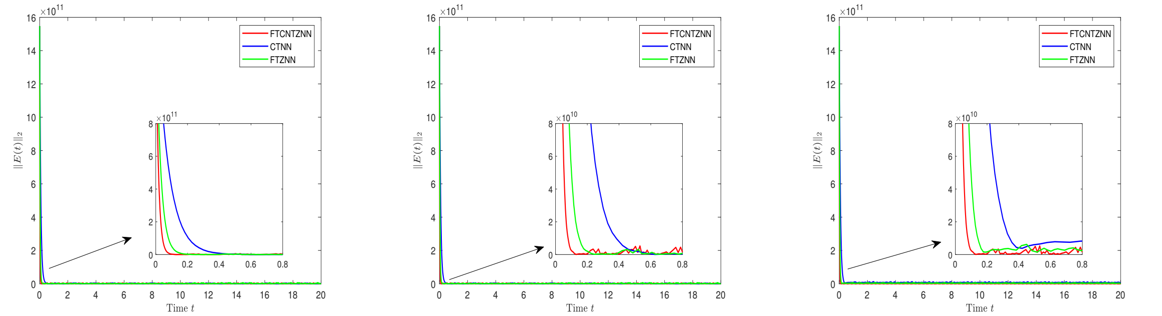



(a) m=4, n=10 (b) m=4, n=20 (c) m=4, n=50

**Figure 1.** The comparison of the FTCNTZNN and CTNN, FTZNN for solving Example 5.1.

Secondly, this study further investigates the impact of different noise environments on the convergence performance of three ZNN models (FTCNTZNN, CTNN, and FTZNN) under a fixed tensor structure of order $m = 4$ and dimension $n = 10$. The Figure 2 illustrate the evolution of the error norm $\|E(t)\|_2$ over time under three distinct conditions no noise, dynamic vanishing noise, and dynamic non-vanishing noise.

Under the noise-free condition (Figure 2 (a)), all three methods converge rapidly to zero, with FTCNTZNN achieving the fastest convergence, followed by FTZNN and CTNN. In the presence of dynamic vanishing noise (Figure 2 (b)), where noise amplitude decays over time, the convergence trajectories are perturbed during the early stages, yet all methods eventually




(a) m=4, n=10, with noise-free $z_j(t) = 0$

(b) m=4, n=10, with vanishing noise $z_j(t) = 0.5e_j(t)$

(c) m=4, n=10, with non-vanishing noise $z_j(t) = 10$

**Figure 2.** The comparison of the FTCNTZNN and CTNN, FTZNN with different noise for solving Example 5.1.

approach the same steady-state level as in the noise-free case. FTCNTZNN again demonstrates better noise tolerance and convergence stability.

Under dynamic non-vanishing noise (Figure 2 (c)), the error behavior differs markedly among the three methods: FTCNTZNN still converges to zero, whereas both CTNN and FTZNN fail to converge, exhibiting divergence. This demonstrates that FTCNTZNN possesses significantly stronger robustness and the ability to suppress persistent disturbances, while the other two ZNN variants are unable to stabilize under such conditions.

Finally, the type of noise significantly affects the convergence behavior of ZNN models, vanishing noise only temporarily delays convergence, while non-vanishing noise leads to persistent steady-state residual errors. Across all tested environments, FTCNTZNN consistently exhibits faster convergence and superior disturbance rejection, confirming its favorable applicability for solving tensor systems in noisy settings.

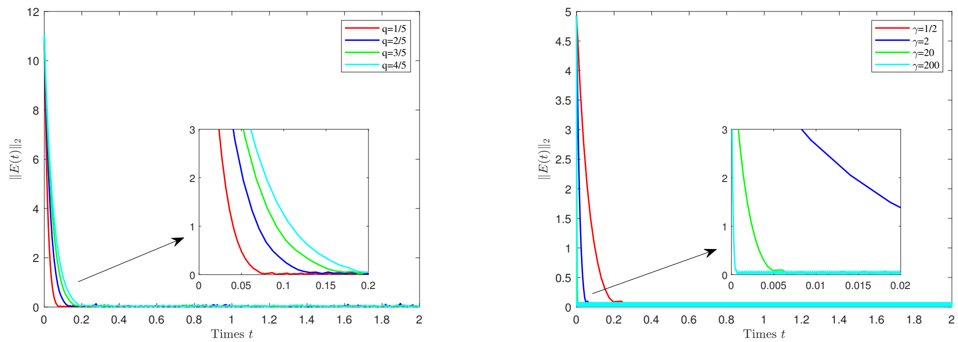
**Example 5.2.** We consider a nonsingular $\mathcal{M}$-tensor $\mathcal{A} = s\mathcal{I} - \mathcal{B}$ [3]. $\mathcal{I}$ denotes an identity tensor with ones along its diagonal and zeros elsewhere. The nonnegative tensor $\mathcal{B} \in \mathbb{R}^{[m,n]}$ comprises elements randomly chosen from the uniform distribution on (0, 1). The scalar $s$ is defined as

$$s = (1+\epsilon) \max_{i=1,2,\ldots,n} (\mathcal{B}e^2)_i, \epsilon > 0,$$

where $e = (1, 1, \ldots, 1)^\top$. In our tests, we choose $b = (1, 2, \ldots, n)^\top$ and the initial value as $x_0 = (2, 2, \ldots, 2)^\top$. In actually computing, we consider the case of $\mathcal{A} \in \mathbb{R}^{[3,n]}$ and $\mathcal{A} \in \mathbb{R}^{[4,20]}$, where $n = 5, 20, 50$.

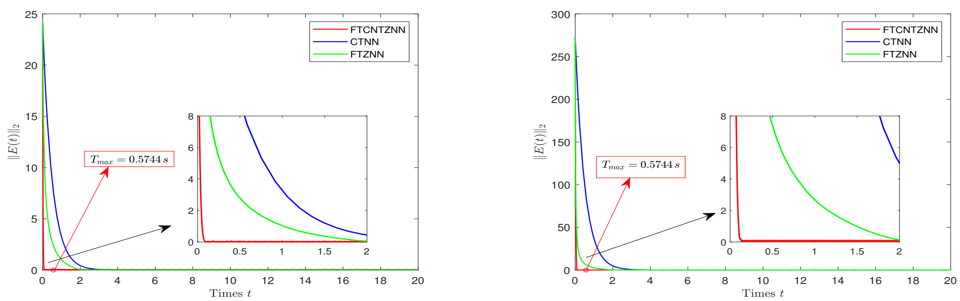
The second example uses the parameters specified in Case 2. Firstly, two types of parameters $q$ and $\gamma$, affect the convergence rate of the FTCNTZNN model (3.5), are shown in Figure 3. From Figure 3 (a) and Figure 3 (b), the comparison of the parameters $q$ and $\gamma$ are in the ideal environment. And we choose the tensor $\mathcal{A} \in \mathbb{R}^{[3,5]}$ as a comparison. Then we define a function $T_{\max}(q)$ on $q$ that describes the upper bound on the FTCNTZNN model's convergence time. According to Figure 3(a), $T_{\max}(\frac{1}{5}) < T_{\max}(\frac{2}{5}) < T_{\max}(\frac{3}{5}) < T_{\max}(\frac{4}{5})$, indicating that the parameter $q$ has a significant impact on the convergence time of the FTCNTZNNN model. Similarly, the parameter $\gamma$ affects the convergence time, as seen in Figure 3(b). Larger $\gamma$ values lead to faster convergence rates and shorter convergence times.

In this paper, we consider the noise tolerance of the FTCNTZNN, CTNN, and FTZNN models under the noise-free environment (i.e., $Z(t) = 0$), the dynamic bounded vanishing noise (i.e., $Z(t) = 0.5E(t)$), and the dynamic bounded non-vanishing noise (i.e., $Z(t) = 0.5$). Secondly,

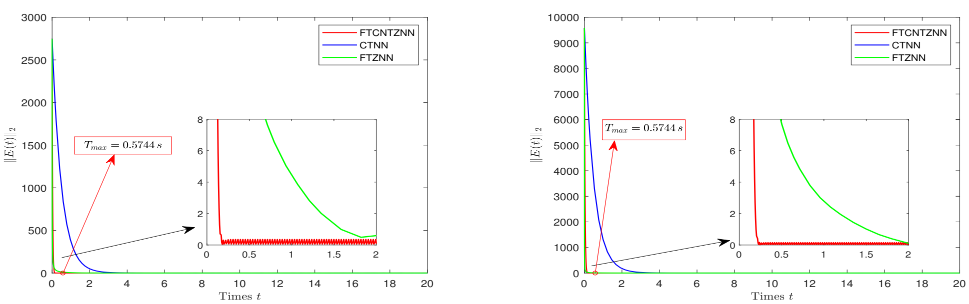



(a) FTCNTZNN model with different parameter q (b) FTCNTZNN model with different parameter $\gamma$

**Figure 3.** m=3, n=5 FTCNTZNN model with different parameters under Case 2.



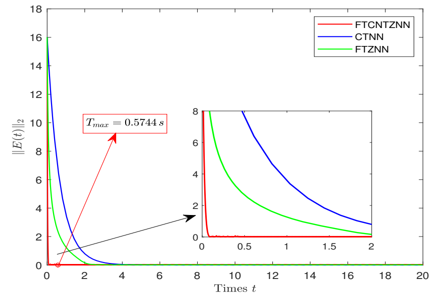

(a) m=3, n=5 (b) m=3, n=20



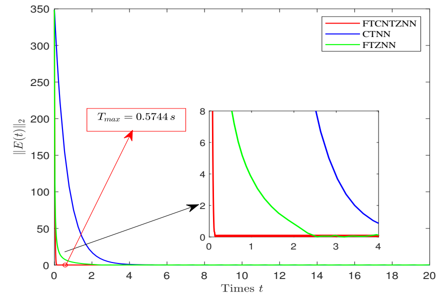

(c) m=3, n=50 (d) m=4, n=20

**Figure 4.** The convergence rate of different models for solving Example 5.2.

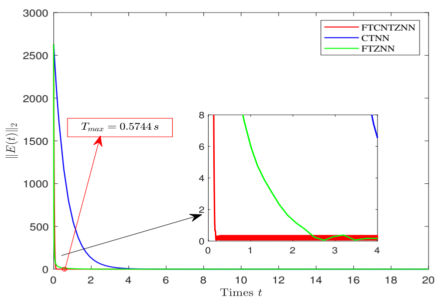
the FTCNTZNN model (3.5) is utilized to solve this example under the noise-free environment with the parameters as specified in Case 2. The appropriate results are presented in Figure 4. It can be seen that the state solution $x(t)$ can converge to the theoretical solution $x_*(t)$ in a fixed time when the error function $\|E(t)\|_2$ can converge to zero. As shown in Figure 4, regardless of the tensor $\mathcal{A} \in \mathbb{R}^{[m,n]}$, the FTCNTZNN model (3.5) converges faster than the CTNN and FTZNN models. We also mean that the FTCNTZNN model can converge within the theoretical fixed time bound $T_{\max} \approx 0.5744s$ however the tensor $\mathcal{A} \in \mathbb{R}^{[m,n]}$ and the environment changes. Figure 5 displays three ZNN models with dynamic bounded vanishing noise (i.e., $z_j(t) = 0.5e_j(t)$) can converge to zero. In Figure 6, we compare the FTCNTZNN model (4.2) with the CTNN and FTZNN models under the environment of dynamic bounded non-vanishing



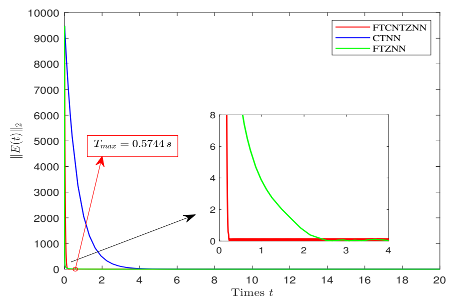

(a) m=3, n=5, FTCNTZNN with $z_j(t) = 0.5e_j(t)$




(b) m=3, n=20, FTCNTZNN with $z_j(t) = 0.5e_j(t)$

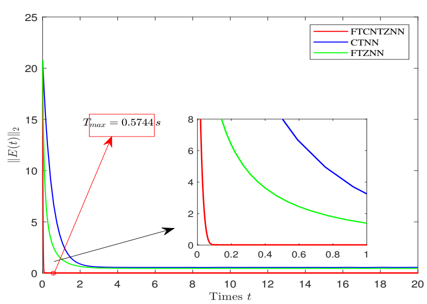



(c) m=3, n=50, FTCNTZNN with $z_j(t) = 0.5e_j(t)$

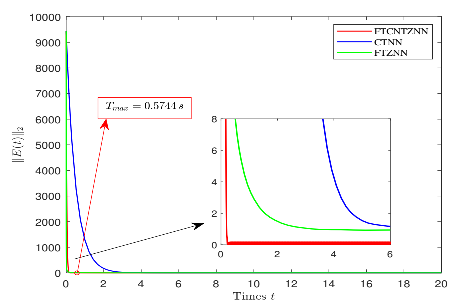



(d) m=4, n=20, FTCNTZNN with $z_j(t) = 0.5e_j(t)$

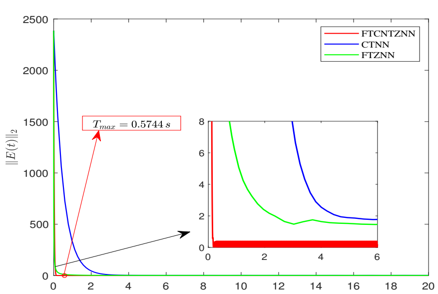
**Figure 5.** The dynamic bounded vanishing noise of different models for solving Example 5.2.



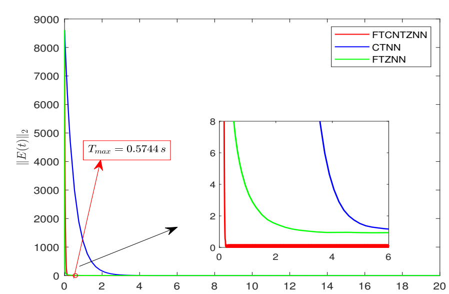

(a) m=3, n=5, FTCNTZNN with $z_j(t) = 0.5$




(b) m=3, n=20, FTCNTZNN with $z_j(t) = 0.5$




(c) m=3, n=50, FTCNTZNN with $z_j(t) = 0.5$




(d) m=4, n=20, FTCNTZNN with $z_j(t) = 0.5$

**Figure 6.** The dynamic bounded non-vanishing noise of different models for solving Example 5.2.

noise (i.e., $z_j(t) = 0.5$). We clearly find only the FTCNTZNN model (4.2) can converge to 0. With the influence of non-vanishing noise, the CTNN and FTZNN models can't converge to zero.

Thirdly, to show the effectiveness and robustness of the FTCNTZNN model (4.2), we compare the graph Figure 4 with Figure 5 and Figure 6. By observing Figure 4 and Figure 5, we know that the dynamic bounded vanishing noise only delays the convergence time of the CTNN and FTZNN models. And it does not affect the convergence of ZNN models. Among them, one can see that the dynamic bounded vanishing noise does not affect the convergence time of the FTCNTZNN model. Comparing Figure 4 with Figure 6, we can get that the influence of dynamic bounded non-vanishing noise on the CTNN and FTZNN models are very large. It makes the state solutions of the CTNN and FTZNN models can't converge to the theoretical solution. However, the FTCNTZNN model (4.2) can still converge to zero. Then we obtain the conclusion that the FTCNTZNN model has strong noise tolerance and the fastest convergence rate.

## 6. Conclusion

This paper proposes a noise-tolerant zeroing neural network model with fixed-time convergence to solve the multi-linear system with $\mathcal{M}$-tensor. Theoretical analysis shows that our proposed noise-tolerant zeroing neural network model with fixed-time convergence model not only maintains noise tolerance, but also has faster convergence time and a convergence time upper bound compared to other zeroing neural network models. Comparing the noise-tolerant zeroing neural network model with fixed-time convergence with the continuous time neural network and finite-time zeroing neural network models, numerical experimental results also confirm the effectiveness and excellent fixed time convergence of the noise-tolerant zeroing neural network model with fixed-time convergence.

## References

[1] M.-Y. Deng, W. Ding and X.-P. Guo, *Modified splitting methods for solving non-homogenous multi-linear equations with $\mathcal{M}$-tensors*, Pacific Journal of Optimization, An International Journal, 2022, 18(1), 91–117.

[2] W. Ding, L. Qi and Y. Wei, *$\mathcal{M}$-tensors and nonsingular $\mathcal{M}$-tensors*, Linear Algebra and its Applications, 2013, 439(10), 3264–3278.

[3] W. Ding and Y. Wei, *Solving multi-linear systems with $\mathcal{M}$-tensors*, Journal of Scientific Computing, 2016, 68(2), 689–715.

[4] H. Duan, X. Xiao, J. Long and Y. Liu, *Tensor alternating least squares grey model and its application to short-term traffic flows*, Applied Soft Computing, 2020, 89, 106145.

[5] C. Han, B. Zheng and J. Xu, *A modified noise-tolerant ZNN model for solving time-varying Sylvester equation with its application to robot manipulator*, Journal of the Franklin Institute, 2023, 360(12), 8633–8650.

[6] L. Han, *A homotopy method for solving multilinear systems with $\mathcal{M}$-tensors*, Applied Mathematics Letters, 2017, 69, 49–54.

[7] H. He, C. Ling, L. Qi and G. Zhou, *A globally and quadratically convergent algorithm for solving multilinear systems with $\mathcal{M}$-tensors*, Journal of Scientific Computing, 2018, 76, 1718–1741.

[8] S. Hu, Z.-H. Huang, C. Ling and L. Qi, *On determinants and eigenvalue theory of tensors*, Journal of Symbolic Computation, 2013, 50, 508–531.

[9] H. Huang, D. Fu, J. Zhang, X. Xiao, G. Wang and S. Liao, *Modified Newton integration neural algorithm for solving the multi-linear $\mathcal{M}$-tensor equation*, Applied Soft Computing, 2020, 96, 106674.

[10] J. Jin, J. Zhu, L. Zhao and L. Chen, *A fixed-time convergent and noise-tolerant zeroing neural network for online solution of time-varying matrix inversion*, Applied Soft Computing, 2022, 130, 109691.

[11] H. K. Khalil, *Nonlinear Systems*, Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[12] D.-H. Li, H.-B. Guan and X.-Z. Wang, *Finding a nonnegative solution to an $\mathcal{M}$-tensor equation*, Pacific Journal of Optimization, An International Journal, 2020, 16, 419–440.

[13] D.-H. Li, S. Xie and H.-R. Xu, *Splitting methods for tensor equations*, Numerical Linear Algebra with Applications, 2017, 24(5), e2102.

[14] W. Li, D.-D. Liu and S.-W. Vong, *Comparison results for splitting iterations for solving multi-linear systems*, Applied Numerical Mathematics, 2018, 134, 105–121.

[15] X. Li and M. K. Ng, *Solving sparse non-negative tensor equations: Algorithms and applications*, Frontiers of Mathematics in China, 2015, 10(3), 649–680.

[16] D. Liu, W. Li and S.-W. Vong, *The tensor splitting with application to solve multi-linear systems*, Journal of Computational and Applied Mathematics, 2018, 330, 75–94.

[17] M. Liu, H. Wu and M. Shang, *A noise-suppressing discrete-time neural dynamics model for solving time-dependent multi-linear $\mathcal{M}$-tensor equation*, Neurocomputing, 2023, 520, 240–249.

[18] Z. Luo, L. Qi and N. Xiu, *The sparsest solutions to $\mathcal{Z}$-tensor complementarity problems*, Optimization Letters, 2017, 11(3), 471–482.

[19] C.-Q. Lv and C.-F. Ma, *A Levenberg-Marquardt method for solving semi-symmetric tensor equations*, Journal of Computational and Applied Mathematics, 2018, 332, 13–25.

[20] M. Najafi Kalyani, M. Nobakht Kooshkghazi and L. Reichel, *Block iterative methods for solving multi-linear systems*, Numerical Algorithms, 2025, 100, 803–829.

[21] M. Nobakht-Kooshkghazi and M. Najafi-Kalyani, *Improving the Gauss-Seidel iterative method for solving multi-linear systems with $\mathcal{M}$-tensors*, Japan Journal of Industrial and Applied Mathematics, 2024, 41, 1061–1077.

[22] A. Polyakov, *Nonlinear feedback design for fixed-time stabilization of linear control systems*, IEEE Transactions on Automatic Control, 2012, 57(8), 2106–2110.

[23] L. Qi, *Eigenvalues of a real supersymmetric tensor*, Journal of Symbolic Computation, 2005, 40(6), 1302–1324.

[24] S. Wang, L. Jin, X. Du and P. S. Stanimirovi, *Accelerated convergent zeroing neurodynamics models for solving multi-linear systems with $\mathcal{M}$-tensors*, Neurocomputing, 2021, 458, 271–283.

[25] X. Wang, M. Che and Y. Wei, *Neural network approach for solving nonsingular multi-linear tensor systems*, Journal of Computational and Applied Mathematics, 2020, 368, 112569.

[26] X. Wang, M. Che and Y. Wei, *Neural networks based approach solving multi-linear systems with $\mathcal{M}$-tensors*, Neurocomputing, 2019, 351, 33–42.

[27] X. Wang, C. Mo, S. Qiao and Y. Wei, *Predefined-time convergent neural networks for solving the time-varying nonsingular multi-linear tensor equations*, Neurocomputing, 2022, 472, 68–84.

[28] H. Wu, S. Wang, X. Du and M. Liu, *Discrete-time recurrent neural network for solving multi-linear $\mathcal{M}$-tensor equation*, Advances in Intelligent Systems and Computing, 2022, 1409, 131–143.

[29] L. Xiao and B. Liao, *A convergence-accelerated Zhang neural network and its solution application to Lyapunov equation*, Neurocomputing, 2016, 193, 213–218.

[30] K. Xie and S.-X. Miao, *A new preconditioner for Gauss-Seidel method for solving multilinear systems*, Japan Journal of Industrial and Applied Mathematics, 2023, 41, 1159–1173.

[31] Z.-J. Xie, X. Jin and Y. Wei, *Tensor methods for solving symmetric $\mathcal{M}$-tensor systems*, Journal of Scientific Computing, 2018, 74, 412–425.

[32] D.-X. Yan, C.-Q. Li, J.-Y. Wu, et al., *A novel error-based adaptive feedback zeroing neural network for solving time-varying quadratic programming problems*, Mathematics, 2024, 12(13), 2090.

[33] H. Zhang and L. Wan, *Zeroing neural network methods for solving the Yang-Baxter-like matrix equation*, Neurocomputing, 2020, 383, 409–418.

[34] Y. Zhang, D. Jiang and J. Wang, *A recurrent neural network for solving Sylvester equation with time-varying coefficients*, IEEE Transactions on Neural Networks, 2002, 13(5), 1053–1063.

Received September 2025; Accepted January 2026; Available online August 2026.